Deep Learning

Project Report

Image Classification using CNN Deep Networks

M.Tech CSE

Team Members:

1. Sidhant Moza (23303026)
2. Ehtesham Ashraf (23003021)
3. Himani Agrawal (23303025)

Table of Contents

Abstract

In recent years, image classification has garnered significant attention in the field of computer vision and machine learning. This paper presents a deep learning approach for classifying images into six categories: 'buildings', 'forest', 'glacier', 'mountain', 'sea', and 'street'. A Convolutional Neural Network (CNN) model was employed, leveraging its exceptional ability to extract hierarchical features from raw image data. The dataset was preprocessed by resizing and normalization, and subsequently split into training, validation, and test sets. The CNN model was then trained on the training and validation data, achieving a remarkable 85% accuracy on the hold-out test dataset. The results demonstrate the effectiveness of deep CNN models in tackling complex image classification problems and pave the way for further research and applications in computer vision and related domains.

Introduction

Image classification, a fundamental task in computer vision, aims to assign predefined labels or categories to digital images. With the rapid advancement of deep learning techniques, particularly Convolutional Neural Networks (CNNs), significant progress has been made in the field of image classification. CNNs excel at automatically learning hierarchical representations and extracting relevant features from raw image data, enabling accurate classification across diverse domains.

Traditional image classification approaches relied on hand-crafted features and shallow machine learning models, which often struggled to capture the complexity and nuances present in natural images. However, the advent of deep learning has revolutionized the field, allowing for end-to-end training of models directly from pixel data, without the need for manual feature engineering.

Several deep CNN architectures, such as AlexNet, VGGNet, ResNet, and Inception, have achieved remarkable performance on various image classification benchmarks. These models typically consist of multiple convolutional layers, pooling layers, and fully connected layers, enabling the extraction of increasingly abstract and discriminative features from the input images.

The performance of image classification models is influenced by several factors, including the quality and diversity of the training data, the model architecture, the optimization technique employed during training, and the computational resources available. Data augmentation techniques, such as rotation, flipping, and color jittering, can help improve model generalization and robustness.

Accurately classifying natural images, encompassing diverse categories like buildings, forests, glaciers, mountains, seas, and streets, is a challenging yet crucial task with numerous

practical applications. It plays a vital role in areas such as remote sensing, environmental monitoring, urban planning, and content-based image retrieval systems.

In industry, image classification is widely utilized in various domains, including e-commerce (product categorization), healthcare (medical image analysis), surveillance (object detection and recognition), and social media (content moderation and organization). As the demand for intelligent systems capable of understanding and interpreting visual data continues to grow, advancements in image classification techniques will play a pivotal role in enabling more sophisticated and intelligent applications.

Literature Review

Paper	Model	Dataset	Remarks
[1] Automatic Convolutional Neural Architecture Search for Image Classification Under Different Scenes	● CNN ● CNAS, ● Reinforcement learning, ● Gradient-Based Optimization,	It has 100 classes containing 600 images each, total 60000 images	● The best architecture achieved higher performance than Darts (a differential architecture search method) across all image datasets used in the study.
[2] RepConv: A novel architecture for image scene classification on Intel scenes.	● ReSNet, ● ReSNet 50, ● ReSNet 101, ● RepConv	The data contains around 25k images of size 150x150 distributed under 6 categories (building; forest; glacier; mountain; sea; street).	● For the multi-classification task, the proposed RepConv model achieves accuracies of 93.55% ± 0.11% and 75.54% ± 0.14% on training and validation data, respectively. ● In the binary classification problem (natural scenes vs. real scenes), the model

			achieves accuracies of 98.08% ± 0.05% on training data and 92.70% ± 0.08% on validation data.
[3] Evolving and Ensembling Deep CNN Architectures for Image Classification.	● Block-wise CNNs, ● Evolutionary algorithms, ● Particle Swarm Optimization, ● Fine-Tuning and Ensembling	CIFAR-10 dataset is used 32x32 image size, 60000 images, 45,000 for training, 5000 for validation and 10000 for testing.	● The error rate was reduced from 4.66% to 4.27% with the first approach. ● The error rate was reduced from 4.66% to 4.39% with the second approach.
[4] Plant Classification using Convolutional Neural Networks.	● CNN, ● ReLU Activation Function, ● Stochastic Gradient Descent ● SVM ● LBP and GIST	Dataset is collected through TARBIL project, of 4800 images across 16 plant classes.	● The CNN-based approach achieved an impressive accuracy of approximately 97.47% in classifying sixteen different plant species, outperforming SVM classifiers using LBP and GIST features.
[5] Transfer Learning for Image Classification.	● VGG 16, ● VGG 19, ● AlexNet, ● Transfer Learning, ● SVM	ILSVRC database consists of 22000 categories of objects.	● The study also investigates the robustness of CNN features by employing support vector machine (SVM) for image classification and comparing its performance with the discussed CNN networks. ● Results indicate a significant improvement

			in the average recall, precision, and F-score when using the VGG19 CNN architecture compared to AlexNet and VGG16, suggesting its superiority in image classification tasks.
[6] Places: An Image Database for Deep Scene Understanding	● AlexNet ● GoogLeNet ● VGG16 ● ResNet152 ● ImageNet-CNN with linear SVM for baseline comparison	About 60 million images (color images of at least 200 200 pixels size). Also used Places205 dataset with 2.5 million images and Places88 dataset with 88 classes datasets. Created benchmark dataset Places365 with 1.8 M images	It highlights the importance of these datasets in addressing currently intractable visual recognition problems, such as determining actions in environments, spotting inconsistent objects or human behaviors, and predicting future events based on scenes.

Methodology

The image classification process was done in Python language on Google Colab Platform.

Dataset

The dataset used in this project is 'Intel Image Classification Dataset' obtained from Kaggle. It consisted of images belonging to six classes: buildings, forests, glaciers, mountains, seas, and streets. Out of the total of approximately 24000 images in the original dataset (training set + validation set + testing set), we used 7,200 images for training, 1,800 images for validation and a hold-out set of 3000 images for testing. The training and validation sets were used for model training and hyperparameter tuning, while the hold-out test set was used for evaluating the final model performance.

Data Preprocessing

The dataset was preprocessed to ensure consistent image dimensions and normalization of pixel values. Each image was resized to 128x128 pixels, and the pixel values were normalized to the range of 0 to 1 by dividing by 255.

Model Architecture

The CNN model architecture was designed and implemented using the TensorFlow and Keras libraries in Python. The model consisted of multiple convolutional layers, max-pooling layers, dropout layers, and dense layers. The specific architecture of the model is shown in the table below as the model's summary.

S.No.	Layer (type)	Output Shape	# Parameters
1	Conv2D	(None, 126, 126, 16)	448
2	Conv2D	(None, 124, 124, 16)	2320
3	MaxPooling2D	(None, 62, 62, 16)	0
4	Conv2D	(None, 60, 60, 32)	4640
5	Conv2D	(None, 58, 58, 32)	9248
6	Dropout	(None, 58, 58, 32)	0
7	MaxPooling2D	(None, 29, 29, 32)	0
8	Conv2D	(None, 27, 27, 32)	18496
9	Conv2D	(None, 25, 25, 32)	36928
10	Dropout	(None, 25, 25, 32)	0
11	MaxPooling2D	(None, 12, 12, 32)	0
12	GlobalAveragePooling2D	(None, 64)	0
13	Dense	(None, 32)	2080
14	Dense	(None, 32)	1056
15	Dense	(None, 6)	198

1. Convolutional layers with 16, 32, and 64 filters, respectively, followed by ReLU activation and max-pooling layers.

2. Dropout layers with a rate of 0.5 to prevent overfitting.
3. A global average pooling layer to reduce the spatial dimensions.
4. Two dense layers with 32 units and ReLU activation.
5. A final dense layer with 6 units and softmax activation for multi-class classification.

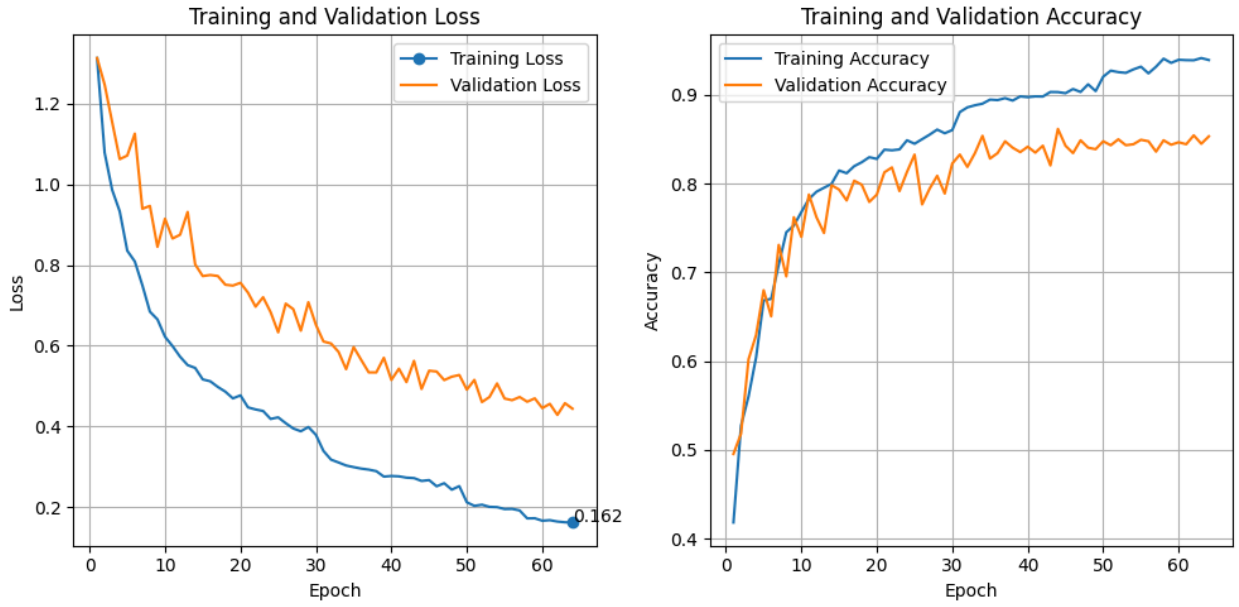
The model was compiled with the Adam optimizer and sparse categorical cross-entropy loss function. The validation accuracy metric was used to evaluate the model's performance during training and validation.

Model Training

The CNN model was trained using the preprocessed training and validation datasets. The training process involved optimizing the model's weights and biases to minimize the loss function and maximize the accuracy on the validation set. Techniques such as early stopping with patience 20, learning rate reduction on plateau by half with patience 5, and model checkpointing were employed to improve the model's performance and prevent overfitting. The initial maximum epoch setting was 150 epochs but the training was stopped at 64 epochs by early stopping to prevent any overfitting of the model.

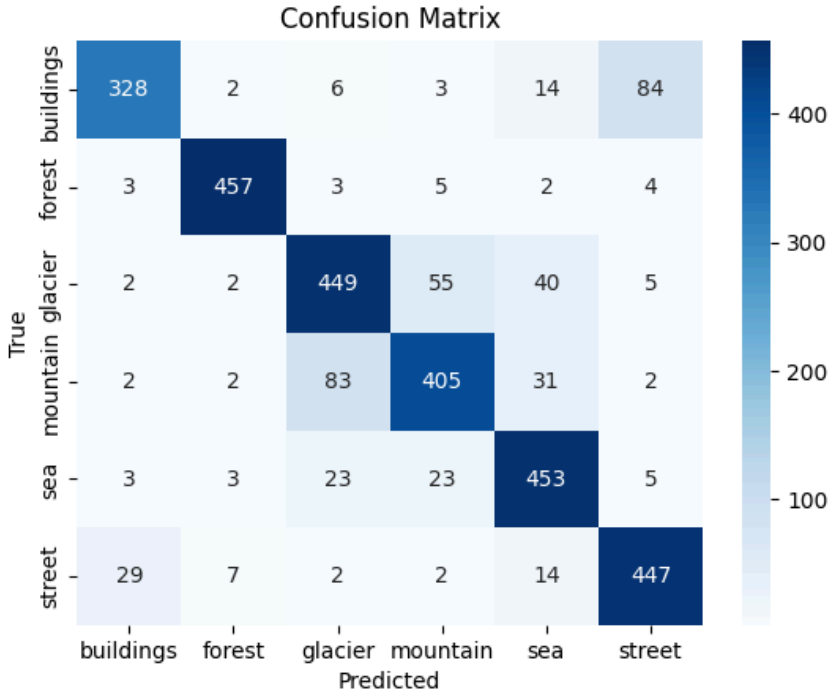
Results

At the end of training and validation, training accuracy was 93.9%, validation accuracy was 86.2%, training loss was 0.16 and validation loss was 0.44. The training and validation loss and accuracy curves are shown below:



Since there is no major difference between training and validation results and since the model training was stopped before validation loss could increase, the model has not overfitted to the data, and has given good results.

Furthermore, the model was saved, and then tested on a hold-out test dataset consisting of 3000 images. The model achieved an accuracy of 84.63% and an F1-score of 84.73%. The confusion matrix and classification report were generated to provide detailed insights into the model's performance across different classes, as shown below:



The classification report was generated as follows:

	Precision	Recall	F1-Score	Support
Buildings	0.89	0.75	0.82	437
Forest	0.97	0.96	0.97	474
Glacier	0.79	0.81	0.80	553
Mountain	0.82	0.77	0.80	525
Sea	0.82	0.89	0.85	510
Street	0.82	0.89	0.85	501

	Precision	Recall	F1-Score	Support
Buildings	0.89	0.75	0.82	437
Forest	0.97	0.96	0.97	474
Glacier	0.79	0.81	0.80	553
Mountain	0.82	0.77	0.80	525
Accuracy			0.85	3000
Macro-Avg	0.85	0.85	0.85	3000
Weighted-Avg	0.85	0.85	0.85	3000

The confusion matrix and the classification report revealed that the model performed well in classifying forests, seas, and streets, with high true positive rates. However, it exhibited some confusion between buildings, glaciers, and mountains, which contributed to the lower precision and recall scores for these classes.

The classification report provided class-wise precision, recall, and F1-scores, along with overall accuracy and macro/weighted averages. The report highlighted that the model performed reasonably well across all classes, with forests having the highest F1-score of 0.97 and glaciers having the lowest F1-score of 0.80.

Conclusion

The CNN model trained on the Intel Image Classification Dataset demonstrated good generalization, achieving an accuracy of 84.63% and an F1-score of 84.73% on the hold-out test set. Employing techniques such as early stopping, learning rate reduction, and model checkpointing ensured effective prevention of overfitting, with training accuracy at 93.9% and validation accuracy at 86.2%. Despite its overall success, the model exhibited difficulty distinguishing between buildings, glaciers, and mountains, leading to lower precision and recall scores in these classes. However, it excelled in classifying forests, seas, and streets, with forests boasting the highest F1-score of 0.97. This indicates the model's robust performance across most classes, highlighting its capability in image classification tasks. Moving forward, further refinement and potentially augmenting the dataset could improve classification accuracy, particularly for challenging classes.

References

[1] Weng Y, Zhou T, Liu L, Xia C. Automatic convolutional neural architecture search for image classification under different scenes. IEEE access. 2019 Mar 28;7:38495-506.

[2] Soudy M, Afify Y, Badr N. RepConv: A novel architecture for image scene classification on Intel scenes dataset. International Journal of Intelligent Computing and Information Sciences. 2022 May 1;22(2):63-73.

[3] Fielding B, Lawrence T, Zhang L. Evolving and ensembling deep CNN architectures for image classification. In2019 International Joint Conference on Neural Networks (IJCNN) 2019 Jul 14 (pp. 1-8). IEEE.

[4] Yalcin H, Razavi S. Plant classification using convolutional neural networks. In2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics) 2016 Jul 18 (pp. 1-5). IEEE.

[5] Shaha M, Pawar M. Transfer learning for image classification. In2018 second international conference on electronics, communication and aerospace technology (ICECA) 2018 Mar 29 (pp. 656-660). IEEE.

[6] Zhou B, Khosla A, Lapedriza A, Torralba A, Oliva A. Places: An image database for deep scene understanding. arXiv preprint arXiv:1610.02055. 2016 Oct 6.